Technical Note:
GPAS Governance Namespace
Code List Component Description

Version 2.1, 2011-07-11

GWG ASFE FG Chair, NGA/NCGIS, asfechair@nga.mil
Paul Birkel, MITRE, pbirkel@mitre.org

1 Introduction

This paper briefly describes the code list component of the Geospatial Publicly Available Specifications (GPAS) governance namespace in the Metadata Registry (MDR) component of the Department of Defense (DoD) Data Services Environment. The purpose of the GPAS code list component is to provide configuration-controlled copies of publicly available geospatial-related code lists that are available through the Internet. These lists of codes and their corresponding specifications are established as individual dictionaries.

By providing these documents on the MDR, operations which otherwise do not have access to the Internet (*e.g.,* Disconnected, Intermittent, and Low-bandwidth (DIL) or enclave environments) will be able to access these code list dictionaries through copies of the MDR hosted locally or on [Internet] disconnected networks.

In addition, the GPAS code list component provides a repository for publicly available lists that is not dependent upon non-DoD assets.

The content of code lists established in the GPAS is exactly as specified by the corresponding standard; in some cases the quality of the definition associated with the code in the standard may be limited – or there may be no additional information in the standard regarding the code beyond a “bare name”.

Further information regarding the GPAS governance namespace is available at:

http://metadata.ces.mil/mdr/ns/GPAS

2 Code Lists

The values of domains whose allowed values can be completely listed are termed **enumerations** and may be established both as sets, and as (contained) individual domain value items.

A **code list** is a flexible enumeration that specifies an unambiguous identifier using a consistent representation for each member in a set of domain values (*e.g.,* country codes or units of measure).

A **code list dictionary** specifies the domain members of a single code list.

When a code list value is instantiated in an XML instance document, two identifiers are provided:

codeList: specifies a remote resource in which the code list domain is defined; and

codeListValue: the coded value from the specified (remote resource code list) domain that encodes the concept intended (denoted) in the XML instance document.

ISO 19136:2007 Geography Markup Language (GML), Clause 15, specifies a schema for dictionaries of item definitions and their identifiers. This XML schema is used to document code lists in the MDR.

GPAS code lists are represented as individual dictionaries – each establishing a unique *codeList* identifier that is recognized as a persistent resource in the MDR.

The members of a code list are represented as individual items/definitions – each establishing a unique *codeListValue* that is also recognized as a persistent resource in the MDR.

3 Code List Resources

3.1 Dictionary of Code Lists

All code list dictionaries established by the GPAS Governance Namespace are accessible through a single "root" **dictionary of code list dictionaries** resource that is accessible at:

http://metadata.ces.mil/mdr/ns/GPAS/codelist

This resource is an XML instance document conformant to a *gml:Dictionary*. It has an associated XSL stylesheet that when applied results in a suitable presentation of the primary dictionary content in a web browser.

Figure 1 illustrates a sample presentation as seen when the (above) specified resource URL is accessed.

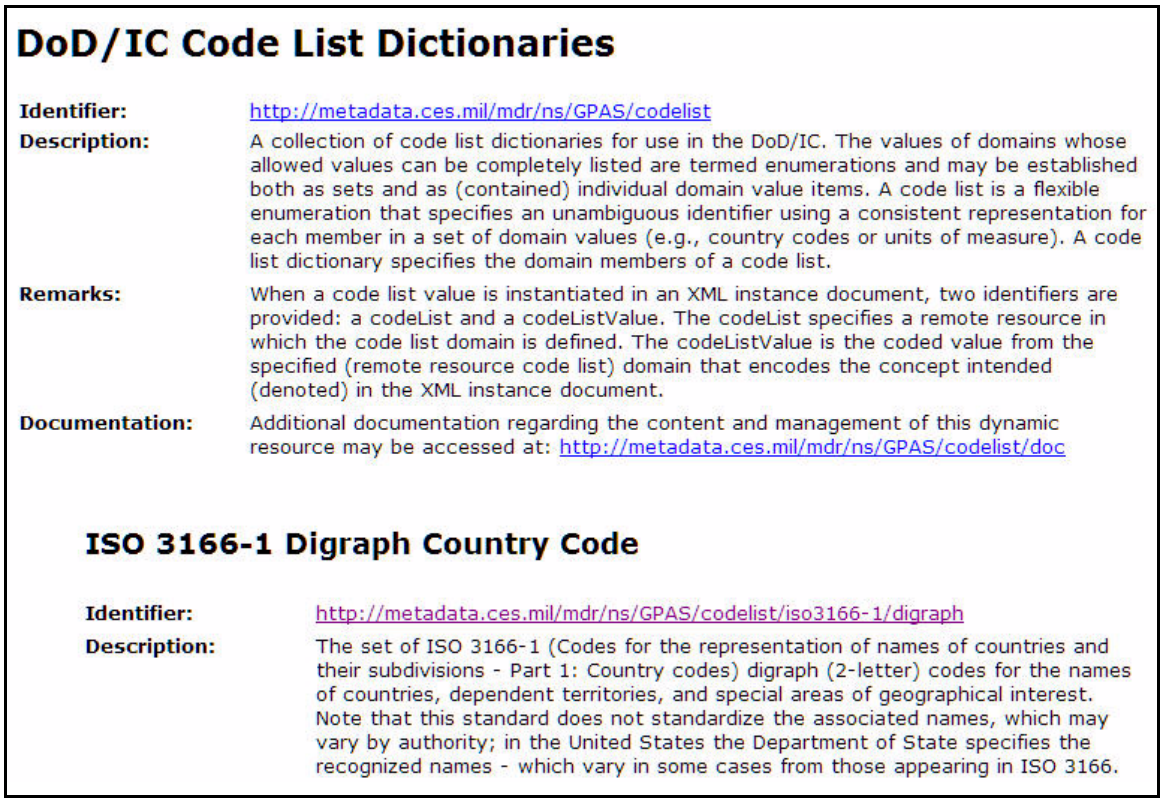


Figure 1 – Sample Presentation of a Dictionary of Code List Dictionaries

Figure 2 illustrates a corresponding sample XML instance document as downloaded from the MDR resource URL.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="gmlDictionary.xsl" type="text/xsl"?>
<gml:Dictionary
   xmlns:gml="http://www.opengis.net/gml/3.2"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gml/3.2
                       http://metadata.ces.mil/mdr/ns/GPAS/schemas/ogc/gml/3.2.1/gml.xsd"
   gml:id="CodelistDictionaries">
```

```
<!-- ======== Version 1.0, 11/29/2007; as of 14-Nov-09 9:06:44 AM ======== -->
<gml:description>……...</gml:description>
<gml:identifier codeSpace="http://metadata.ces.mil/mdr/ns/GPAS">codelist</gml:identifier>
<gml:name>DoD/IC Code List Dictionaries</gml:name>
<gml:remarks>……...</gml:remarks>

<gml:dictionaryEntry>
  <gml:Dictionary gml:id="Iso3166-1_Digraph">
    <gml:description>……...</gml:description>
    <gml:identifier
      codeSpace="http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-1">digraph</gml:identifier>
    <gml:name>ISO 3166-1 Digraph Country Code</gml:name>
  </gml:Dictionary>
</gml:dictionaryEntry>
…
</gml:Dictionary>
```

Figure 2 – Sample XML instance document for a Dictionary of Code List Dictionaries

3.2 Code List Dictionary

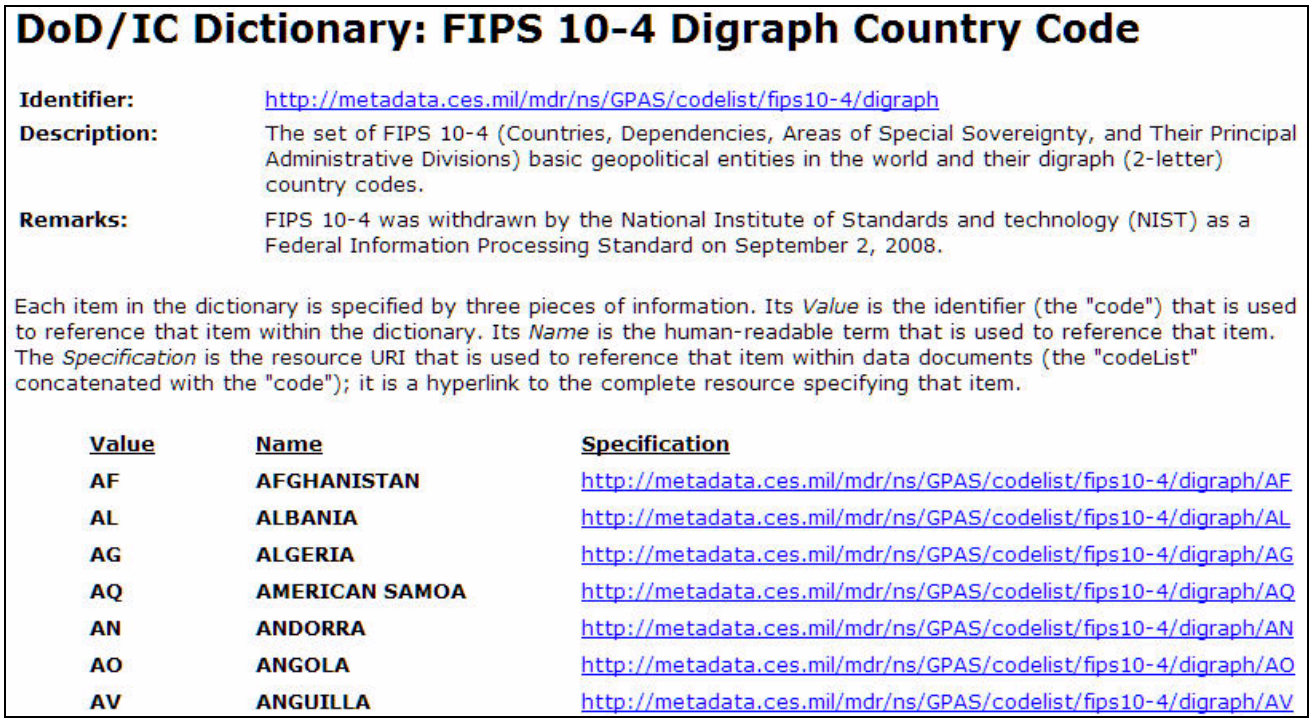
Individual code list dictionaries are established as persistent resources within the MDR using a relative path expression from the **GPAS/codelist** dictionary-of-dictionaries root. Table 1 lists a few of these code list resources.

Table 1 - Example GPAS Code List Resources

Code List	Resource URL
FIPS 10-4 Digraph	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph
FIPS 10-4 Division	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/division
IANA Charset	http://metadata.ces.mil/mdr/ns/GPAS/codelist/ianaCharset
ISO 3166-1 Digraph	http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-1/digraph
ISO 3166-1 Trigraph	http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-1/trigraph
ISO 3166-2	http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-2
ISO 639-2	http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso639-2

Each resource is an XML instance document conformant to a *gml:Dictionary*. It has an associated XSL stylesheet that when applied results in a suitable presentation of the primary dictionary content in a web browser.

Figure 3 illustrates a sample presentation as seen when the "**ISO 3166-1 Digraph**" resource URL is accessed. Following the presentation of the essential specification of the Code List is a two-column table listing each allowed Code List Value in terms of its: (1) Domain Value (a *gml:Definition/gml:identifier*), (2) Name (a *gml:Definition/gml:name*), and (3) accompanying Resource URL (the concatenation of the values of *gml:Definition/gml:identifier/@codespace*, '*/*', and *gml:Definition/gml:identifier*).



DoD/IC Dictionary: FIPS 10-4 Digraph Country Code

Identifier: http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph

Description: The set of FIPS 10-4 (Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Divisions) basic geopolitical entities in the world and their digraph (2-letter) country codes.

Remarks: FIPS 10-4 was withdrawn by the National Institute of Standards and technology (NIST) as a Federal Information Processing Standard on September 2, 2008.

Each item in the dictionary is specified by three pieces of information. Its *Value* is the identifier (the "code") that is used to reference that item within the dictionary. Its *Name* is the human-readable term that is used to reference that item. The *Specification* is the resource URI that is used to reference that item within data documents (the "codeList" concatenated with the "code"); it is a hyperlink to the complete resource specifying that item.

Value	Name	Specification
AF	AFGHANISTAN	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph/AF
AL	ALBANIA	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph/AL
AG	ALGERIA	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph/AG
AQ	AMERICAN SAMOA	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph/AQ
AN	ANDORRA	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph/AN
AO	ANGOLA	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph/AO
AV	ANGUILLA	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph/AV

Figure 3 – Sample Presentation of a Code List Dictionary

The dictionary XML instance document may be downloaded for local use and contains the complete specification of all of its *codeListValues*.

Figure 4 illustrates a corresponding sample XML instance document as downloaded from the MDR resource URL, including just one of its multiple *gml:DictionaryEntry* values (*i.e.,* "AFGHANISTAN").

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="../gmlDictionary.xsl" type="text/xsl"?>
<gml:Dictionary
   xmlns:gml="http://www.opengis.net/gml/3.2"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/gml/3.2
                       http://metadata.ces.mil/mdr/ns/GPAS/schemas/ogc/gml/3.2.1/gml.xsd"
   gml:id="Fips10-4_Digraph">

   <!-- ======== Version 1.0, 11/29/2007; as of 14-Nov-09 9:06:43 AM ======== -->
   <gml:description>The set of FIPS 10-4 (Countries, Dependencies, Areas of Special Sovereignty, and
Their Principal Administrative Divisions) basic geopolitical entities in the world and their digraph (2-letter)
country codes.</gml:description>
   <gml:identifier
     codeSpace="http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4">digraph</gml:identifier>
   <gml:name>DoD/IC Dictionary: FIPS 10-4 Digraph Country Code</gml:name>

   <gml:dictionaryEntry>
      <gml:Definition gml:id="AF">
         <gml:description>[unspecified]</gml:description>
         <gml:identifier
           codeSpace="http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph">AF</gml:identifier>
         <gml:name>AFGHANISTAN</gml:name>
      </gml:Definition>
   </gml:dictionaryEntry>
   ...
```

```
</gml:Dictionary>
```

Figure 4 – Sample XML instance document for a Code List Dictionary

3.3 Code (in a Code List)

Individual code list items (*codeListValues*) are established as persistent resources within the MDR using a relative path expression from the code list-specific root. Table 2 illustrates a few of these. Note that as the Resource URL establishes identity it may be used for comparison and equality-testing locally without necessarily being dereferenced with the corresponding resource retrieved and examined.

Table 2 - Example GPAS Code List Resources

Code	Resource URL
ISO 3166-1 Digraph value for **Argentina**	http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-1/digraph/AR
ISO 3166-1 Trigraph value for **Australia**	http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-1/trigraph/AUS
ISO 3166-2 value for **Delvine, Armenia**	http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-2/AL-DL
FIPS 10-4 Digraph value for **Andorra**	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/digraph/AN
FIPS 10-4 Division value for **Texas (USA)**	http://metadata.ces.mil/mdr/ns/GPAS/codelist/fips10-4/division/US48
ISO 639-2 value for **Armenian**	http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso639-2/arm
IANA Charset value for **UTF-8**	http://metadata.ces.mil/mdr/ns/GPAS/codelist/ianaCharset/UTF-8

Each resource is an XML instance document conformant to a *gml:Definition*. It has an associated XSL stylesheet that when applied results in a suitable presentation of the complete item content in a web browser.

Figure 5 illustrates a sample presentation as seen when the "**ISO 3166-1 Digraph**" Specification (resource URL) for the Domain Value "**ANTIGUA AND BARBUDA**" is accessed.

Figure 5 – Sample Presentation of a Code Specification

The corresponding XML instance document may be downloaded for local use and contains the complete specification of the *codeListValue*.

Figure 6 illustrates the corresponding sample XML instance document as downloaded from the MDR resource URL.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="../../../gmlDefinition.xsl" type="text/xsl"?>
<gml:Definition
   xmlns:gml="http://www.opengis.net/gml/3.2"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
    xsi:schemaLocation="http://www.opengis.net/gml/3.2
                        http://metadata.ces.mil/mdr/ns/GPAS/schemas/ogc/gml/3.2.1/gml.xsd"
    gml:id="AG">
    <!-- ======== Version 1.0, 11/29/2007; as of 14-Nov-09 9:06:44 AM ======== -->
    <gml:description>[unspecified]</gml:description>
    <gml:identifier
       codeSpace="http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-1/digraph">AG</gml:identifier>
    <gml:name>ANTIGUA AND BARBUDA</gml:name>
    <gml:remarks>Includes Redonda Island</gml:remarks>
</gml:Definition>
```

Figure 6 – Sample XML instance document for a Code

4 Code List Resource Use

4.1 CodeListValueType

As an example application of code list resources, define the complex type *CodeListValueType* as follows:[1]

```
<complexType name="CodeListValueType">
  <simpleContent>
    <extension base="string">
      <attribute name="codeList" type="anyURI" use="required"/>
      <attribute name="codeListValue" type="string" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

The XML Schema *xs:anyURI* type is a finite-length character sequence which, when the algorithm defined in Section 5.4 of [XML Linking Language] is applied to it, results in a string that is a legal URI according to [RFC 2396], as amended by [RFC 2732].[2]

This *CodeListValueType* type may be substituted for the type of any element that would otherwise be an *xs:string*.

For example, an XML schema might contain an element declaration using this type:

```
<element name="country" type=" CodeListValueType "/>
```

An element corresponding to this might appear in an XML instance document as follows:

```
<country  codeList="http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso3166-1/trigraph"
          codeListValue="USA"/>
```

4.2 Schematron Constraints

ISO/IEC 19757-3:2006 defines the Schematron Document Schema Definition Language (DSDL) that may be used to specify one or more validation processes to be performed against XML instance documents. Schematron is a rule-based validation language for making assertions about the presence or absence of patterns in XML trees. It is a simple and powerful structural schema language expressed in XML using a small number of elements and XPath (a query language for selecting nodes from an XML document). It may be employed as an adjunct to the structural validation capabilities of XSD – testing for co-occurrence constraints, non-regular constraints, and inter-document constraints.

---

[1] This type pattern is used in both ISO 19136 (GML) and ISO/TS 19139 Metadata encoding). A similar pattern is employed in the DoD Discovery Metadata Specification (DDMS), there specified using the XML attributes *qualifier* and *value*.

[2] Note that spaces are, in principle, allowed in values of *anyURI*, however, their use is highly discouraged (unless they are encoded by '%20').

Schematron is a language system for specifying and declaring assertions about arbitrary patterns in XML documents, based on the presence or absence, names and values of elements and attributes along paths. It uses the languages of *XML Path Language (XPath) Version 1.0* and *XSL Transformations (XSLT) Version 1.0.*

- Considered as a document type, a Schematron schema (*.sch* file) contains natural-language assertions concerning a set of XML documents, marked up with various elements and attributes for testing these natural-language assertions, and for simplifying and grouping assertions.
- Considered theoretically, a Schematron schema reduces to a non-chaining rule system whose terms are Boolean functions invoking an external query language on the instance and other visible XML documents, with syntactic features to reduce specification size and to allow efficient implementation.
- Considered analytically, Schematron has two characteristic high-level abstractions: the pattern and the phase. These allow the representation of non-regular, non-sequential constraints that ISO/IEC 19757-2:2003 (*Document Schema Definition Languages (DSDL) – Part 2: Regular grammar-based validation – RELAX NG*) cannot specify, and various dynamic or contingent constraints.

A general Schematron validator is a function returning an indication that an XML document is "valid", "invalid" or "error". The function notionally performs two steps: transforming the specified Schematron schema into a minimal syntax, and then testing the XML document against the minimal syntax. It is common to implement Schematron validators directly using XSLT.

The W3C XML Schema recommendations allow applications to extend schema validation by adding application specific data in *appinfo* elements within the *annotation* of a particular schema element. One can embed *sch:pattern* elements within these extension blocks, which can then be applied as part of the schema validation process. Namespaces that are used by patterns are declared in an *annotation* at the top level of the schema using *ns* elements.

A prototypical schema specification using such an embedding is illustrated in Figure 7, where the single test specifies that only three particular values of the *codeListValue* are allowable – and it is assumed that these values are well-known since the *codeList* is specified uniquely as “MyCodeSpace”.

```
<element name="country" type="CodeListValueType">
  <annotation>
    <documentation>Country: A simple codelist-based specification of an ISO 3166-1
        3-character code limited to one of three values in a codelist of “convenience”.</documentation>
    <appinfo>
      <sch:pattern id="countryAllowableCode">
        <sch:rule context="ns:country">
          <sch:assert test="(@codeList = ‘MyCodeSpace’)
                and (@codeListValue =’USA’ or @codeListValue =’CAN’ or @codeListValue =’MEX’)"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
```

Figure 7 – Sample XML Schema for a Country Element

A valid element corresponding to this might appear in an XML instance document as follows:

```
<country codeList="MyCodeSpace" codeListValue=”USA”/>
```

Note that the use of an ISO/IEC 19757:2006 Schematron assertion is specified “in line” in the schema where it would need to be extracted for use in a Schematron validator. This assertion may instead be placed in a separate “*sch*” file for greater ease-of-use in processing XML instance documents.

4.3 MDR-based Code List Specification

Instead of assuming the use of well-known values for the XML attribute *codeListValue* and not-necessarily well-known values for the *codeList*, as illustrated in Figure 7, developers may instead use the identifiers of code lists and their domain members as registered in the MDR.

Specification of code list domains “by reference” in the MDR-hosted code list information resource allows for dynamic support for allowable domain values independent of evolution of the domain-specific schema itself. This ensures that code list domain values in XML instance documents can be validated against publicly-visible, shared codelist-specific sets of values. Schematron assertions may be used to assert additional constraints on the allowed code list values for individual uses in a domain-specific schema.

Figure 8 illustrates a simple schema specification for the element *ns:language* whose type is *CodeListValueType* as specified in Section 4.2.

```
<element name="language" type="CodeListValueType">
  <annotation>
    <documentation>Language: An ISO 639-2 (Codes for the representation of names of languages - Part 2: Alpha-3 code) trigraph (3-letter) code for the representation of the names of languages.</documentation>
  </annotation>
</element>
```

Figure 8 – Sample XML Schema for a Language Element

An element corresponding to this might appear in an XML instance document as follows:

<language codeList="'http://metadata.ces.mil/mdr/ns/GPAS/codelist/iso639-2" codeListValue=”eng”/>

Figure 9 illustrates a portion of a Schematron document intended to accompany the XSD specifying the use of XML element *ns:language*; it specifies the GPAS resource URL and then uses it to specify a series of assertions grouped into a single rule to test instances of the *ns:language* element in an XML instance document.

```
<sch:let name="GPAS" value="'http://metadata.ces.mil/mdr/ns/GPAS'"/>

<sch:pattern id="LanguageCode_Valid_in_Resource">
  <sch:rule context="ns:language">

    <sch:assert test="@codeList = concat($GPAS, '/codelist/iso639-2')">
        The code list is registered in the MDR GPAS Governance Namespace.</sch:assert>

    <sch:let name="url" value="concat(@codeList, '/', @codeListValue)"/>
    <sch:assert test="document($url)">
        The URL '<sch:value-of select="$url"/>' must reference a net-accessible resource.</sch:assert>

    <sch:assert test="normalize-space(.) = ''">The element must be empty.</sch:assert>

    <!-- Verify that the content of the resource matches the instance document -->
    <sch:assert test="@codeList = document($url)/gml:Definition/gml:identifier/@codeSpace">
        The specified codeList must match the codeSpace of the identifier in the resource.</sch:assert>

    <sch:assert test="@codeListValue = document($url)/gml:Definition/gml:identifier">
        The specified codeListValue must match the value of the identifier in the resource.</sch:assert>

  </sch:rule>
</sch:pattern>
```

Figure 9 – Sample SCH Document for a Language Element

Five *sch:assert* are specified, as follows:

1. The value of the *codeList* attribute is validated as using the recognized **ISO 639-2** code list from the DoD/IC Code List Dictionaries. This is accomplished by a local syntactic examination of the attribute value.
2. The URL constructed from the *codeList* and *codeListValue* attributes is validated as using a currently-recognized code list member of the **ISO 639-2** code list from the DoD/IC Code

List Dictionaries. This is accomplished by determining whether the URL references a net-accessible resource.

3. The element value is verified as being unpopulated.[3]
4. The net-accessible resource is validated as specifying the indicated code list.
5. The net-accessible resource is validated as specifying the indicated code list value.

The second *sch:assert* is not strictly necessary as if it fails then both the fourth and fifth *sch:assert* will also fail.

If it can be assured (and thus assumed) that the resource specified by $GPAS (the MDR in the present case) shall only contain valid information resources then the fourth and fifth *sch:assert* are superfluous and may be dropped; the presence of the resource itself is sufficient.

[3] If the *CodeListValueType* were to have not been defined as an extension of *xs:string* this assertion would not be necessary/desirable.